- In the video, First steps in computer vision, Laurence Maroney introduces us to the Fashion MNIST data set and using it to train a neural network in order to teach a computer “how to see.” One of the first steps towards this goal is splitting the data into two groups, a set of training images and training labels and then also a set of test images and test labels. Why is this done? What is the purpose of splitting the data into a training set and a test set?
    - The training dataset shows the computer that a certain pattern is to be matched with a certain label. The testing dataset contains similar patterns that the computer has not seen before, and therefore the testing dataset can show how well the computer has learned to recognize these patterns.

- The fashion MNIST example has increased the number of layers in our neural network from 1 in the past example, now to 3. The last two are .Dense layers that have activation arguments using the relu and softmax functions. What is the purpose of each of these functions. Also, why are there 10 neurons in the third and last layer in the neural network.
    - The relu function is an activation function (encapsulated math processes) that sets negative neuron output values to zero. The purpose is to eliminate meaningless negative outputs so they do not affect positive outputs in other places.
    - Softmax function is also an activation function, it finds the most likely candidate among the labels. Softmax sets the largest value to 1 and the rest to 0, and then the computer just needs to find the 1.
    - There are 10 neurons in the last layer to calculate the probability that each item has to belong to 1 of the 10 classes

- In the past example we used the optimizer and loss function, while in this one we are using the function adam in the optimizer argument and sparse_categorical-crossentropy for the loss argument. How do the optimizer and loss functions operate to produce model parameters (estimates) within the model.compile() function?
    - The loss function estimates the percent of correct and incorrect guesses in the model, while the optimizer adjusts the parameters to make a better guess

- Using the mnist drawings dataset (the dataset with the hand written numbers with corresponding labels) answer the following questions.
    - What is the shape of the images training set (how many and the dimension of each)?
        - 28x28
        - There are 60,000
    - What is the length of the labels training set?
        - 60,000
    - What is the shape of the images test set?
        - 10,000, 28x28

Estimate a probability model and apply it to the test set in order to produce the array of probabilities that a randomly selected image is each of the possible numeric outcomes (look towards the end of the basic image classification exercises for how to do this — you can apply the same method applied to the Fashion MNIST dataset but now apply it to the hand written letters MNIST dataset).

- Use np.argmax() with your predictions object to return the numeral with the highest probability from the test labels dataset.

  It is 2

- Produce a plot of your selected image and the accompanying histogram that illustrates the probability of that image being the selected number

I chose number 77, and the graph is indeed correct: